1. Pros and Cons: Use Both.

2. Inhibition is also an Important Bias.

### Biology Says: Both

<table>
<thead>
<tr>
<th>Minus Phase</th>
<th>Plus Phase</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_i^+, y_j^+ \approx 0$</td>
<td>$x_i^+, y_j^+ \approx 1$</td>
<td></td>
</tr>
<tr>
<td>$x_i^-, y_j^- \approx 0$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_i^-, y_j^- \approx 1$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Err</th>
<th>Hebb</th>
<th>Combo</th>
<th>Err</th>
<th>Hebb</th>
<th>Combo</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_i^-, y_j^- \approx 0$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>$x_i^-, y_j^- \approx 1$</td>
<td>−</td>
<td>0</td>
<td>−</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

No $Ca^{2+}$ → no learning
Mod $Ca^{2+}$ → LTD
High $Ca^{2+}$ → LTP

![Graph showing LTP and LTD transitions with Ca²⁺ concentration](image)
Error-driven is based on remote errors
Hebbian is local

<table>
<thead>
<tr>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hebbian (local)</td>
<td>autonomous, reliable</td>
</tr>
<tr>
<td></td>
<td>myopic, greedy</td>
</tr>
<tr>
<td>Error-driven (remote)</td>
<td>task-driven, cooperative</td>
</tr>
<tr>
<td></td>
<td>co-dependent, lazy</td>
</tr>
</tbody>
</table>

Error-driven = Left-wing, Hebbian = Right-wing
Combining Error-driven + Hebbian

Get benefits of both:

$$\Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err}$$

(1)
Combining Error-driven + Hebbian

Get benefits of both:

\[ \Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err} \]  \hspace{1cm} (1)

\[ \Delta_{hebb} = \epsilon a_j (a_i - w_{ij}) \]
Combining Error-driven + Hebbian

Get benefits of both:

\[ \Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err} \quad (1) \]

\[ \Delta_{hebb} = \epsilon a_j (a_i - w_{ij}) \]

\[ \Delta_{err} = \epsilon [(a_i^+ a_j^+) - (a_i^- a_j^-)] \]
Combining Error-driven + Hebbian

Get benefits of both:

\[ \Delta w_{ij} \approx \Delta_{hebb} + \Delta_{err} \quad (1) \]

\[ \Delta_{hebb} = \epsilon a_j (a_i - w_{ij}) \]

\[ \Delta_{err} = \epsilon [(a_i^+ a_j^+) - (a_i^- a_j^-)] \]

\[ \Delta w_{ij} = (k_{hebb}) \Delta_{hebb} + (1 - k_{hebb})(\Delta_{err}) \quad (2) \]
Inhibitory Competition as a Bias

Inhibition:

- Causes sparse, distributed representations (many alternatives, only a few relevant at any time).

- Competition and specialization: survival of fittest.

- Self-organizing learning.

(Often more important than Hebbian bias)
The Whole Enchilada

1. Biological realism

2. Distributed Representations

3. Inhibitory Competition

4. Bidirectional Activation Propagation

5. Error-driven Learning

6. Hebbian Learning

\[ \Delta w = \delta_j a_i + \alpha_i \alpha_j \]
Generalization

How well do we deal with things we’ve never seen before?
Generalization

How well do we deal with things we’ve never seen before?

nust
Generalization

How well do we deal with things we’ve never seen before?

nust
Generalization

How well do we deal with things we’ve never seen before?

Each time you walk into class, each social interaction, each sentence you hear, etc.
Generalization

How well do we deal with things we’ve never seen before?

Each time you walk into class, each social interaction, each sentence you hear, etc.

We’re constantly faced with new situations, and generalize reasonably well to them.
Generalization

How well do we deal with things we’ve never seen before?

Each time you walk into class, each social interaction, each sentence you hear, etc.

We’re constantly faced with new situations, and generalize reasonably well to them.

How do we do it?
Generalization

Distributed reps: novel items are novel combinations of existing features (combinatorial representations): “nust”

Hebbian & inhibition: produce elemental, combinatorial reps.
Sims: Generalization
Deep Networks

Need many hidden layers to achieve many stages of transformations (dramatically re-representing the problem).

But then the error signals are very remote & weak.

Need to add constraints and self-organizing learning:

- **a)** Gyrosopes
- **b)** Flexibility Limits

![Diagram showing constraints and self-organizing learning](image_url)
Example: Family Trees

Christo=Penny
Marge=Art
Vicky=James
Jenn=Chuck

Andy=Christi
Colin
Rob=Maria
Charlot
Pierro=Francy

Gina=Emilio
Lucia=Marco
Angela=Tomaso

Alf
Sophia

Agent
Relation
Patient
Sequential & Temporally-Delayed Learning

1. The Problem.

2. Sequential Learning & Context.

3. Temporally-delayed Learning & Reinforcement.
The Problem

Currently: networks learn *immediate* consequence of a given input.

What if current input only makes sense as part of a temporally-extended *sequence* of inputs?

What if the consequence of this input comes *later* in time?
How do we do it?
For example:
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
The pig raced around the pen.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
The pig raced around the pen.

We represent the context, not just the current input.
Sequence Learning

How do we do it?
For example:

My favorite color is purple.
Purple my color favorite is.
Is my purple color favorite.
Is purple my color favorite.

The girl picked up the pen.
The pig raced around the pen.

We represent the context, not just the current input.
in language, social interactions, driving (who goes at a 4-way stop?)
Representing Context for Sequence Learning

How does the brain do it?
How would we get our models to do it?
Representing Context for Sequence Learning

How does the brain do it?
How would we get our models to do it?

Add layers to keep track of context (prefrontal cortex).
Representing Context for Sequence Learning

Simple Recurrent Network (SRN; Elman, Jordan).
Simple Recurrent Network (SRN; Elman, Jordan).

\[ c_j(t) \approx h_j(t - 1) + c_j(t - 1) \]
Simple Recurrent Network (SRN; Elman, Jordan).

\[ c_j(t) \approx h_j(t - 1) + c_j(t - 1) \]

\[ c_j(t) = f_{m_{hid}}h_j(t - 1) + f_{m_{prv}}c_j(t - 1) \]
An Example Task

BTXSE
BPVPSE
BTSXXTVVE
BPTVPSE
An Example Task

BTXSE
BPVPSE
BTSXXTVVE
BPTVPSE

Which of the following sequences are allowed?:
BTXXTTVVE
TSXSE
VVSXE
BSSXSE
An Example Task

BTXSE, BPVPSE, BTSXXTVVE, BPTVPSE, BTXXTTVVE
TSXSE, VVSXE, BSSXSE
An Example Task

BTXSE, BPVPSE, BTSXXTVVE, BPTVPSE, BTXXTTVVE
TSXSE, VVSXE, BSSXSE
An Example Task

BTXSE, BPVPSE, BTSXXTVVE, BPTVPSE, BTXXTTVVE
TSXSE, VVSXE, BSSXSE

We implicitly learn such grammars (e.g., pressing buttons faster to letters that follow grammar).
Randomly chooses one of two possible next states.

Hidden/context units learn to encode states, not labels.
Currently: networks learn *immediate* consequence of a given input.

What if current input only makes sense as part of a temporally-extended *sequence* of inputs? (*context*)

What if the consequence of this input comes *later* in time?
Reinforcement often delayed from the action(s) that lead to it: need to “span the gap”.
Temporally-delayed Learning & Reinforcement

Reinforcement often delayed from the action(s) that lead to it: need to “span the gap”.

Key ideas: We want to predict rewards consistently over time. This process leads us to learn what events are associated with rewards, earlier and earlier back in time.

We use the Temporal Differences (TD) algorithm (Sutton).
Reinforcement Biology

Midbrain dopaminergic (DA) systems modulate cortex & basal ganglia: Substantia Nigra (SN) & Ventral-Tegmental Area (VTA).

SN & VTA are controlled by other cortical/BG areas.

These other areas are like an "Adaptive Critic" (AC), which evaluates stimuli & actions for their rewarding value..
Actual Recordings from Actual Neurons

VTA firing moves from responding to reward to *anticipating* it at the instruction.
Value function, sum of discounted future rewards:

\[ V(t) = \langle \gamma^0 r(t) + \gamma^1 r(t + 1) + \gamma^2 r(t + 2) \ldots \rangle \]  \hspace{1cm} (3)

Recursive definition:

\[ V(t) = \langle r(t) + \gamma V(t + 1) \rangle \]  \hspace{1cm} (4)

Error in predicted reward:

\[ \delta(t) = \left( r(t) + \gamma \hat{V}(t + 1) \right) - \hat{V}(t) \]  \hspace{1cm} (5)
Network Implementation

\[ V(t) = V(t+1) + \delta(t) - r(t) \]
Model: CS at t=2, US at t=16
Phase-based Implementation

Plus phase: AC settles via weights = expected reward at t+1 (or r).

Minus phase: AC clamped to previous plus phase value (0 at start).

Learning goes "backwards in time" to affect previous time step.
CSC input (Complete Serial Compound): a unique unit for each stimulus at each time point.

Not realistic, but good for demonstration.